MKR——一种有效的增量式概念获取系统*

阮俊杰

(中国人民解放军军事科学院军事运筹分析研究所,北京 100091)

摘要 本文对示例式学习的ID5R算法进行了深入的研究并作了进一步的改进,提高了其计算效率.并提出一种适合于增量式学习的抗干扰机制,拓广了算法的适用范围.

关键词 知识获取,机器学习.

ID3[1]算法是示例式学习的一种有效算法,但它却是非增量式的.Fisher的ID4[4]算法,Utgoff的ID5[5]及ID5R[7]算法,均为ID3算法在增量式方面的改进.ID4算法不能保证学习过程的完备性,ID5算法生成的判定树一般较ID3的判定树复杂,ID5R算法虽能生成与ID3算法一致的判定树,但计算费用却很昂贵.在我们开发的MKR系统中,对ID5R算法作了进一步的改进,提高了其计算效率.

目前,很多研究[8,10,11]试图通过对判定树的操作来提高ID3算法的抗干扰能力.在增量式的学习过程中,其适用性受到限制.在我们的MKR系统中,通过将判定树的训练和应用相区别,提出了一种适合于增量式学习的识别与删除噪音的过程.

1 MKR对ID5R判定树生成算法的改进

同ID3算法一样,ID5R算法所基于的示例是通过"属性——值"对的序列来进行描述的.形式地,每个示例均由属性集A中的属性及其取值的序列和类别名称集C中的特定的类别来描述.设当前提供的所有示例构成示例集S.对于A中的任一属性A_i,设其值域为$v_i=\{v_{ij}\}$,则属性A_i相对于示例集S的不确定性或称信息容量可由下面的E-函数或称E-标度来计算:

$$E(A_i,S)=\sum_{j=1}^{|v_i|}\frac{|S|v_{ij}|}{|S|}I(P(S|v_{ij},C_1),\cdots,P(S|v_{ij},C_{|C|})) \tag{1.1}$$

其中:

$$I(X_1,\cdots,X_{|C|})=\begin{cases}0 & \text{当存在正整数 } K_0\leqslant|C|,\text{使 } \sum\limits_{\substack{1\leqslant K\leqslant|C|\\K\neq K_0}}X_K=0\\ -\sum\limits_{K=1}^{|C|}\dfrac{X_K}{\sum\limits_{j=1}^{|C|}X_j}\log_2\dfrac{X_K}{\sum\limits_{j=1}^{|C|}X_j} & \text{其它}\end{cases}$$

---

* 本文1991年3月27日收到,1992年1月6日定稿

作者阮俊杰,33岁,助研,主要研究领域为AI,DB,DSS.

本文通讯联系人:阮俊杰,北京 100091,中国人民解放军军事科学院军事运筹分析研究所

这里,对任何有限集 T,我们用|T|表示其长度.在上式中,$S|v_{ij}$为 S 中满足$[A_i=v_{ij}]$的示例所构成的子集.并设 $C=\{C_1,\cdots,C_{|C|}\}$,用 $P(S|v_{ij},C_k)$表示 $S|v_{ij}$中属于 C_K 类的示例的数量.在 ID3 和 ID5R 的判定树生成算法中,在每一测试结点,选择 E－标度最小的属性作为测试属性.

与 ID3 算法不同,为了适应于增量式的学习,ID5R 算法在其判定树中保存了计算 E－标度所需的必要信息.形式地,ID5R 的判定树由下面两种结点及其连接构成,即:

1. 叶结点,其中含有:

(a)一个类别名.

(b)满足该叶结点路径上测试条件的示例描述集,这种描述由不出现在该叶结点路径上的属性及其取值的序列组成.

2. 非叶结点,称为测试结点或决策结点,含有:

(a)一个测试属性,对于其每一个取值,都对应一个连向其它子树的分枝,并记录该分枝上各类示例的数量.

(b)在该结点路径上所有非测试属性的集合.对其中每一属性的任一取值,记录满足该描述的各类示例的数量.

ID5R 的判定树生成过程由两个算法 Trees－update 和 Pull－up 来实现,可参见第 3 节中表 3.1 和表 3.2 中未标记的语句所构成的算法.在每一次训练后,ID5R 所生成的判定树与基于当前示例集的 ID3 算法所生成的判定树保持一致.

下面,我们来介绍 MKR 对 ID5R 判定树生成算法的改进.为此,先考虑 E－标度的两个性质:

性质 1. 设满足一非叶结点路径上条件的示例构成含有 N 个示例的示例集 S_N,则当新提供一个 C_{K0}类的示例 I 后,如果其满足此结点路径上的条件且在训练时未引起此结点路径上其它结点的测试属性的变化,则对于该结点上的任一测试或非测试属性 A_i,如果在 I 中 A_i 取值 v_{ij0},则有:

$$E(A_i,S_{N+1})=\frac{N}{N+1}E(A_i,S_n)+M(A_i,v_{ij0},S_N) \tag{1.2}$$

其中:

$$M(A_i,v_{ij0},S_N)=-\frac{|S_N|v_{ij0}|}{N+1}I(P(S_N|v_{ij0},C_1),\cdots,P(S_N|v_{ij0},C_{|C|}))+\frac{|S_N|v_{ij0}|+1}{N+1}I(P(S_N|v_{ij0},C_1)),\cdots,P(S_N|v_{ij0},C_{K0}+1,\cdots,P(S_N|v_{ij0},C_{|C|}))$$

$$S_{N+1}=S_N\cup\{I\}$$

限于篇幅,这里省略证明.显然,按公式(1.2)递推地计算新的 E－标度较利用公式(1.1)重新计算要简单得多,特别是当属性的取值较多时.

推论. 在性质 1 的条件下,则有:

$$|E(A_i,S_{N+1})-E(A_i,S_N)|\leqslant\frac{2}{N}$$

性质 2.

$$E(A_i,S|_{[A_{i_1}=v_{i_1j_1}][A_{i_2}=v_{i_2j_2}]\cdots[A_{i_m}=v_{i_mj_m}]})=E(A_i,S|_{T([A_{i_1}=v_{i_1j_1}]\cdots[A_{i_m}=v_{i_mj_m}])})$$

这里 T 为选择子(见[13]中的定义)$[A_{i_1}=V_{i_1j_1}],\cdots,[A_{i_m}=V_{i_mj_m}]$的任一排列函数.

MKR 的判定树,是在 ID5R 判定树的基础上,对于每个非叶结点上的所有测试和非测试属性,还需保存各自的 E－标度值.而且,在每一非叶结点上,还要保存一个状态信息 ST. ST 为一个二维向量,其第一个分量取值 0 或 1,第二个分量的值域为{0,1,…,|C|}.在 MKR 的判定树生成算法中,只涉及其第一个分量 ST(1),其第二个分量的作用在下节介绍.

MKR 的判定树生成算法,是在对 ID5R 的 Trees－update 和 Pull－up 算法进行改进的基础上得出的,对应于第 3 节中表 3.1 和表 3.2 中无标记和标记为"＊"的语句所描述的算法.

在任一已扩展的判定树中,所有非叶结点上的 ST(1)取值均为 0. ST(1)取值 1,当且仅当在判定树的训练过程中,当调用 Pull－up 算法对某一属性进行提升时,递归地,当将该属性提升到上一层某一结点后,将该结点的所有非叶子结点上的 ST(1)置为 1.而在 Trees－update 的算法中,在计算某一结点上各属性的 E－标度时,如果 ST(1)取值为 1,则在修正各属性所有取值上的各类示例数量后,采用公式(1.1)重新计算各属性的 E－标度值,保存这些 E－标度值并将该结点上的 ST(1)值恢复为 0,继续对树的修正过程;如果 ST(1)取值为 0,并且该结点位于满足示例描述的路径上,则先由公式(1.2)递推地计算各属性的 E－标度值,然后修正各属性满足示例描述的取值上的示例数量,保存新计算的各属性的 E－标度值后继续对树的修正过程;而对于其它非叶结点,则终止对其子树的修正过程.

显然,MKR 的判定树生成算法较 ID5R 提高了效率.首先,在很大范围内简化了 E－标度的计算;其次,减少了递归地重构子树的深度或宽度.在最好情况下,计算效果可望提高 b/2 倍,这里 b 为属性值域的平均长度.在最坏情况下,仅能节省判定树根结点上各属性 E－标度的计算.从整体上看,可望 MKR 能够显著地提高 ID5R 算法的效率.然而,MKR 判定树所增加的存储开销相对于 ID5R 判定树本身的存储代价是很小的.

2 MKR 的抗干扰机制

现在,我们假设在前 N＋n 个示例中,存在有 n 个带噪音的训练例.这里,所谓一个示例带有噪音,指这一示例由于噪音的干扰而成为一个错误的命题.我们还假定噪音对各属性取值干扰的随机性是一致的,而且 N 充分大,并有:

$$n/N\leqslant\tau<1(\tau>0\text{ 为一常数}) \tag{2.1}$$

由于 ID5R 从而 MKR 生成的判定树与训练例的顺序无关,因此,我们在考察这 N＋n 个训练例生成的判定树时,不妨先考虑由其中 N 个无噪音训练例生成的判定树 T",然后再考虑 n 个带噪音的训练例对它的影响.

我们设在无干扰情况下所生成的判定树 T"具有稳定的性质,即叶结点的数量与示例总数的比例为一无究小量.则由于 N 充分大,判定树误分类的概率可望充分小[1].对于 n 个带噪音示例集中的任一示例 I,设其描述中的类别名为 $C_{K'_0}$,下面考察根据 I 的属性描述所确定的一个模式按判定树 T"进行分类的情况.由(2.1)式,不妨假设 T"能对这个模式进行正确分类.设这个模式经过 T"的测试满足一个 C_{K_0}类的叶结点路径上的所有测试条件,由于判定

树 T″能正确分类且 I 带有噪音,显然 $K_0 \neq K'_0$. 这时,如果我们在考察 n 个带噪音示例对 T″的干扰时,不再调整判定树 T″的结构,则 I 对 T″进行训练的结果,必将引起 T″中某一 C_{K_0} ($K_0 \neq K'_0$)类叶结点上示例集的进一步划分. 由于我们假设噪音对各属性取值干扰的随机性相同,不难看出在这种特殊的训练方式下,n 个带噪音的训练例完成对 T″的训练后,所生成的判定树 T′具有如下特征:

在判定树 T′的某一子树(与原判定树 T″的一叶结点相对应)中,如将属于 C_K 类的示例数量记为 $|C_K|$,则有正整数 $K_0 \leqslant |C|$,使得:

$$\sum_{\substack{1 \leqslant K \leqslant |C| \\ K \neq K_0}} |C_K| \leqslant \tau |C_{K_0}| \tag{2.2}$$

由(2.1)式,这样的子树显然存在.

下面,对于判定树中的任一测试属性 A_i,定义其在取值 v_{ij} 分枝上的 D−标度如下:

$$D(A_i, v_{ij}) = \left(\sum_{\substack{1 \leqslant K \leqslant |C| \\ K \neq K_0}} |C_K| \leqslant \tau |C_{K_0}| \right) / |C_{K_{max}}| \quad (1 \leqslant j \leqslant |V_i|) \tag{2.3}$$

这里,$|C_K|$ 为在分枝 $[A_i = V_{ij}]$ 上记录的属于 C_K 类的示例数量,K_{max} 由下式确定:

$$|C_{K_{max}}| = \max\{|C_1|, \cdots, |C_{|C|}|\}$$

则对于 T′中具有(2.2)特征的子树,设其父结点上的测试属性为 A_i,且这棵子树对应于 $[A_i = v_{ij}]$ 的分枝,则显然:

$$0 < D(A_i, v_{ij}) \leqslant \tau \tag{2.4}$$

而且,由性质 1 的推论,不难证这棵子树根结点上的测试属性 A_j 的 E−标度满足:

$$0 < E(A_j, S) \leqslant 2\tau \tag{2.5}$$

下面,我们再来考察 n 个带噪音的训练例由 MKR 算法对 T″进行训练实际生成的判定树 T. 由于 n/N 充分小,可望 T 相对于 T′在结构上不会有很大变化. 一般地,由于 T 与 ID3 算法基于同样 N+n 个示例所生成的判定树相同,根据 ID3 算法追求信息熵即 E−标度最小的机制及其有效性,因此 T 一般较 T′为简. 则可望 T 中满足测试属性的 E−标度$\leqslant 2\tau$ 的测试结点的平均深度一般较 T′为小. 对于这样的一个测试结点,设其对应于父结点的 $[A_i = v_{ij}]$ 的分枝,则必然有:

$$D(A_i = v_{ij}) \leqslant \tau$$

即满足(2.4)的特征.

根据示例中存在噪音干扰情况下判定树的特征,我们在 MKR 中开发了一种排除噪音的机制,对应于表 3.1 和表 3.2 中标记为"＊ ＊"的部分.

如果训练例的总数 $N \leqslant \eta$(这里 η 为一小于 1 的阈值),则在对判定树的每一步训练中,对于判定树中除根结点外的每一测试结点,当 MKR 算法确定了其测试属性后,计算其父结点到该结点分枝 $[A_i = V_{ij}]$ 上的 D−标度 $D(A_i, V_{ij})$,如果此 D−标度满足 $D(A_i, V_{ij}) \leqslant \eta$,则将该测试结点上的状态信息 ST(2)标记为 K_{max},否则标记为 0. 在 MKR 的 Pull−up 算法被调用时,对 ST(2)执行与 ST(1)类似的操作(这里为置 0 操作).

为使排除噪音的机制适应于增量式的学习过程,我们将判定树的训练与应用加以区别. 在 MKR 的判定树的分类应用中,一个待分类的模式 I 被分到 C_K 类,当且仅当下面两种情

况之一出现：

1. I 的描述满足某一 C_K 类叶结点路径上的测试条件.

2. 在对 I 的分类路径上，某一结点的 ST(2)标记为 k.

如上所述，这种排除噪音的机制具有其内在的潜力. 在最好的情况下，可望正确地删除所有带噪音的训练例. 在最坏情况下，可能标记为噪音的示例全部为正确示例. 这时，假设当前提供了 m 个训练例，MKR 的判定树共有 L 个叶结点，则根据文[1]的估计，在不考虑删除标记的情况下此判定树错误分类的概率 P 满足：

$$P \leqslant L/2.72m \tag{2.6}$$

由于在最坏情况下最多删除 $\eta \cdot m$ 个示例，则在删除这些示例后判定树错误分类的概率 P_1 满足：

$$P \leqslant L_1/2.72(1-\eta)m \tag{2.7}$$

这里 L_1 为删除操作完成后判定树的叶结点数，显然 $L_1 \leqslant L$.

从(2.6)和(2.7)式，不难看出即使在这种最坏的情况下，MKR 的排除噪音机制对判定树的预测能力并无大的影响，却可带来简化判定树的益处.

3 MKR 的算法描述

如前所述，MKR 算法是在对 ID5R 算法进行改进，并增加了对噪音的检测与删除的有关机制而提出的. 与 ID5R 算法相似，MKR 每一步对判定树的训练是一个对判定树的修正与重构过程. 这个过程可由下面的 Trees—update 算法来实现：

表 3.1 MKR 的 Trees—update 算法

1. 如果判定树为空，则将其置为仅包含一个结点的非扩展形式，将所提供的示例的类别名赋予此结点，其示例集仅由一个示例组成.

2. 否则，如果判定树仍为非扩展形式，并且其类别名与所提供的示例的类别名相同，则将此示例并入该结点的示例集中.

3. 否则，

(a)如果判定树仍为非扩展形式，则将其置为扩展形式，在根结点上任选一个属性作为测试属性. * 置 ST 为(0,0)，将各属性的 E—标度置 0.

(b)在当前测试结点上，如果 ST(1)=1，则根据所提供的训练例，对于当前测试结点上的测试和非测试属性，修正其对应于示例描述中取值的分枝上的有关示例数量，利用公式(1.1)计算各属性的 E—标度值，* 存储这些 E—标度值并将当前结点的 ST(1)置为 0；* 如果 ST(1)=0，则先利用公式(1.2)计算各属性的 E—标度值，存储这些 E—标度值，并修正各属性对应于示例描述中取值分枝上的有关示例数量.

(* *c)如果示例总数 $N \geqslant \frac{1}{\eta}$，则计算父结点到当前结点分枝上的 D—标度，如果此 D—标度值$\leqslant \eta$，则将当前测试结点的 ST(2)置为 K_{max}，否则置为 0.

(d)如果当前测试结点上的测试属性不再具有最小的 E—标度，则：

(i)调用 Pull—up 算法，将具有最小 E—标度的属性 A_i 提升到当前结点并作为测试属性.

(ii)对于除了 A_i 取示例描述中对应值的分枝外的所有分枝，递归地重构它们的直接子树. * 在此过程中，如果某一结点上的 ST(1)取值为 0，则终止对其直接子树的重构过程.

(e)递归地修正判定树中当前结点上 A_i 取示例描述中相应值的分枝下面的子树. 需要的话扩展一个结点，* 并置该

结点的 ST 为(0,0),其中各属性的 E－标度置为 0.

在上面算法中对属性进行提升时,要调用到如下的 Pull－up 算法.

表 3.2　MKR 的 Pull－up 算法

1. 如果属性 A_{new} 已被提升到根结点,终止.

2. 否则,

a)递归地将 A_{new} 提升到其所有直接子树的根结点上.必要时可扩展一个结点,*将该结点的 ST 置为(1,0),其上各属性的 E－标度置为 0.

b)将子树进行变换,生成这样一棵子树,使得 A_{new} 为根结点上的测试属性,其所有直接子树根结点上的测试属性为 Aold,*,**并将这些子树根结点上的 ST 置为(1,0).

4　分析与结论

以下是 MKR 的一个实验性运行的有关结果:

表 4.1　MKR 的一个运行结果

示例数	引起判定树重构的示例数量	带噪音示例数	重新计算 E－标度次数÷计算 E－标度总数	CPU(秒)		η	删除噪音示例数量	删除无噪音示例数量
40	13	12	0.372	ID5R	3.3	5%	0	0
				MKR	1.8			
80	22	12	0.435	ID5R	7.1	5%	0	1
				MKR	4.2			
160	43	12	0.243	ID5R	16.3	5%	4	2
				MKR	7.4			
320	73	12	0.184	ID5R	44.2	5%	11	5
				MKR	12.6			

在这个实验中,我们将 320 个示例随机地提供给 MKR.同时,为便于考察 MKR 删除噪音的效果,前 40 个示例中包含了全部的带噪音的示例.表 4.1 记录了 MKR 在不同运行阶段的有关性能指标.

从 MKR 的运行效率上看,虽然与训练例的输入顺序有很大关系,但在一般情况下 MKR 较 ID5R 显著地提高了效率,如表 4.1 所示的情况.

从 MKR 排除噪音的效果看,只有当提供的示例数量较大时,这种效果才能显著地表现出来.而且,这种效果并不总是与示例的数量呈正比关系.

从 MKR 的抗干扰机制对判定树质量的影响看,首先,噪音示例的剪除,将大大改善判定树的质量.其次,剪除噪音的误操作对判定树的分类效果并无大的影响.第三,噪音示例的剪除操作仅影响到判定树的分类应用,而不影响对它的训练,因此误操作对判定树的影响并不是永久性的.第四,MKR 的剪除操作只是一个标记的过程,未在剪除部分示例后对判定树进行调整.

参考文献

1 Quinlan J R. Learning efficient classification procedures and their applications to chess and games. In: Michalski, Carbonell & Mitchell (eds.), Machine Learning: An Artificial Intelligence Approach, 1983;1:463—482.

2 Quinlan J R. Induction of decision trees. Machine Learning, 1986;1(1):81—106.

3 Quinlan J R. Decision trees as probabilistic classifiers. Proc. of the 4th International Workshop for Machine Learning, 1987:31—37.

4 Schlimmer J C, Fisher O. A case study of incremental concept induction. Proc. of the 5th National Conference on Artificial Intelligence, 495—501.

5 Utgoff P E. ID5: an incremental ID3. Proc. of the 5th International Workshop on Machine Learning, 107—120.

6 Cendrowska J. PRISM: algorithm for inducing modular rules. Man—Machine Studies, 27(4):349—370.

7 Utgoff P E. Incremental induction on decision trees. Machine Learning, 1990;4(2):161—186.

8 Quinlan J R. The effect of noise on concept learning. In: Michalski R *et al*. (eds.), Machine Learning: An Artificial Intelligence Approach, 1986;2,148—166.

9 Utgoff P. Shift of bias for inductive concept learning. In: Michalski R *et al*. (eds.), Machine Learning: An Artificial Intelligence Approach, 1986;2.

10 Quinlan J R. Simplifying decision trees. Man—Machine Studies, 27(3):221—234.

11 Cheng J *et al*. Improved decision trees:a generalized version of ID3. Proc. of the 5th International Workshop on Machine Learning, 100—106.

12 Haussler D. Learning conjunctive concepts in structrual domain. Machine Learning, 1990;4(1):7—41.

MKR——AN EFFICIENT CONCEPTS ACQUISITION SYSTEM

Ruan Junjie

(*Institute of Military Operational Analysis and Research, Academy of Military Science, Beijing* 100091)

Abstract This paper introduces the MKR algorithm for Military Knowledge Acquisition system. MKR is an incremental and efficient algorithm for learning from examples. In MKR, this paper made a further improvement on ID5R algorithm, raising it's efficiency. This paper also developed an anti—disturbance mechanism in MKR, extending the field in which the algorithm can be applied.

Key words Knowledge acquisition, machine learning.